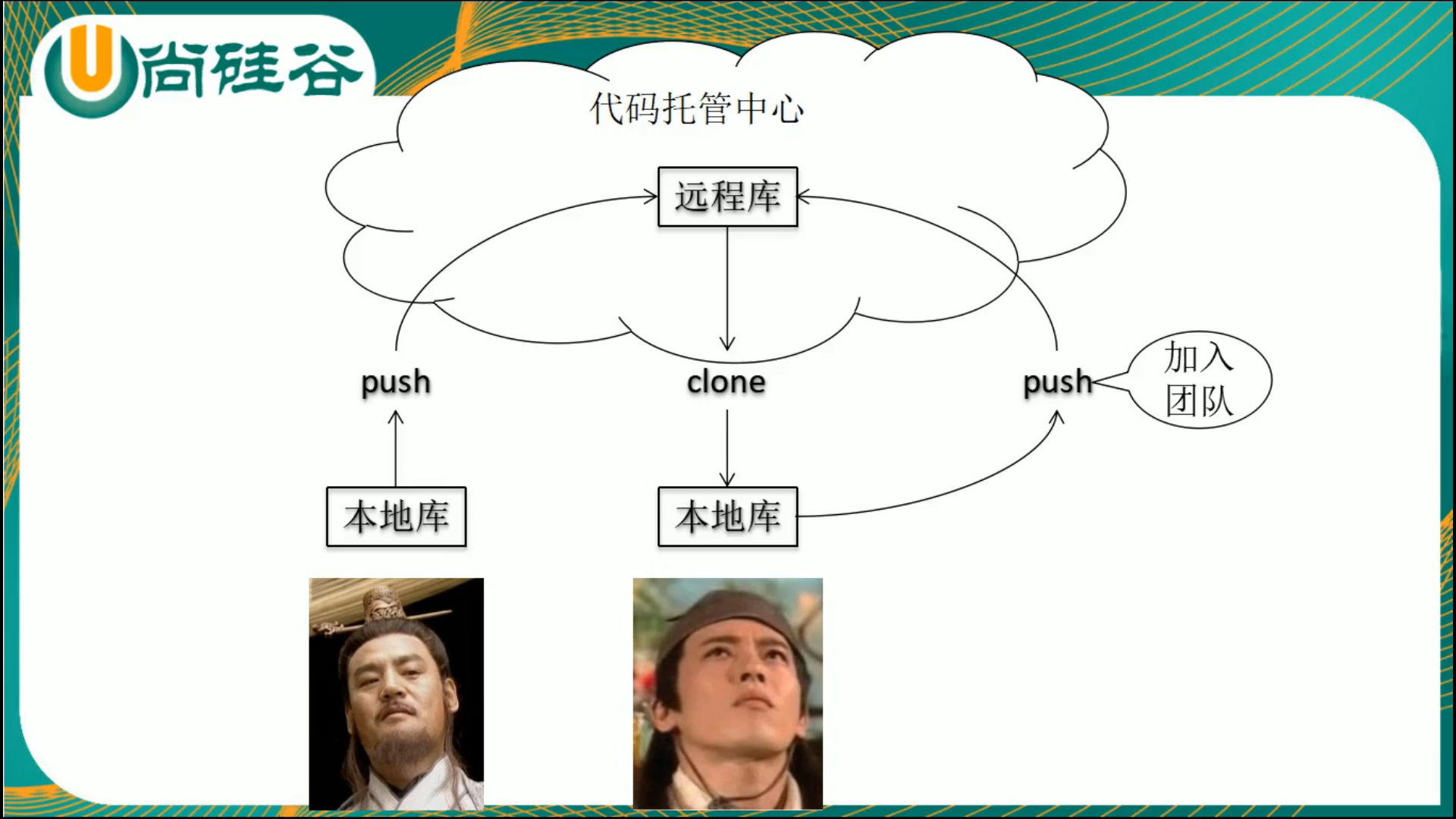
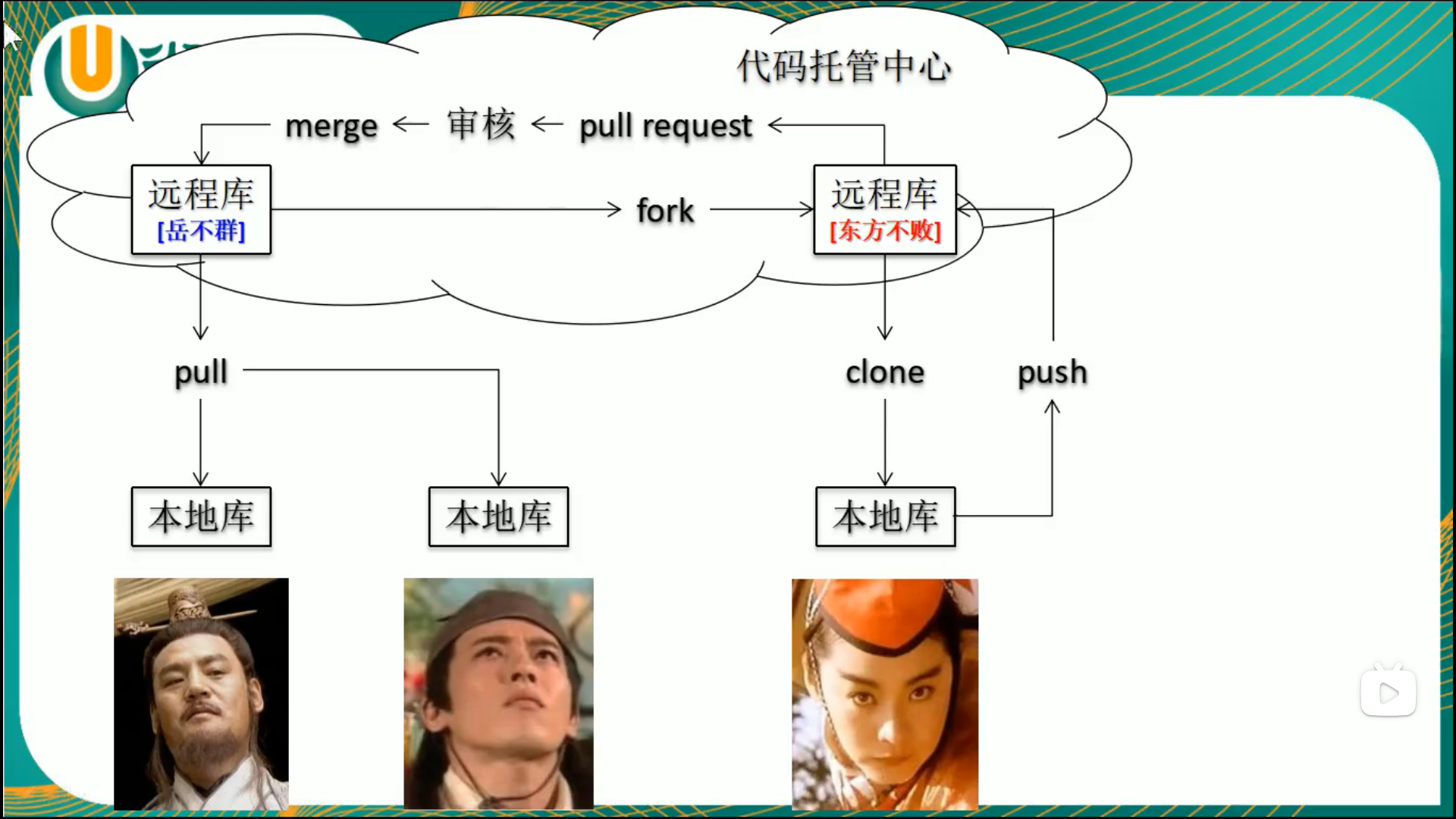
# 1 本地库和远程库交互方式

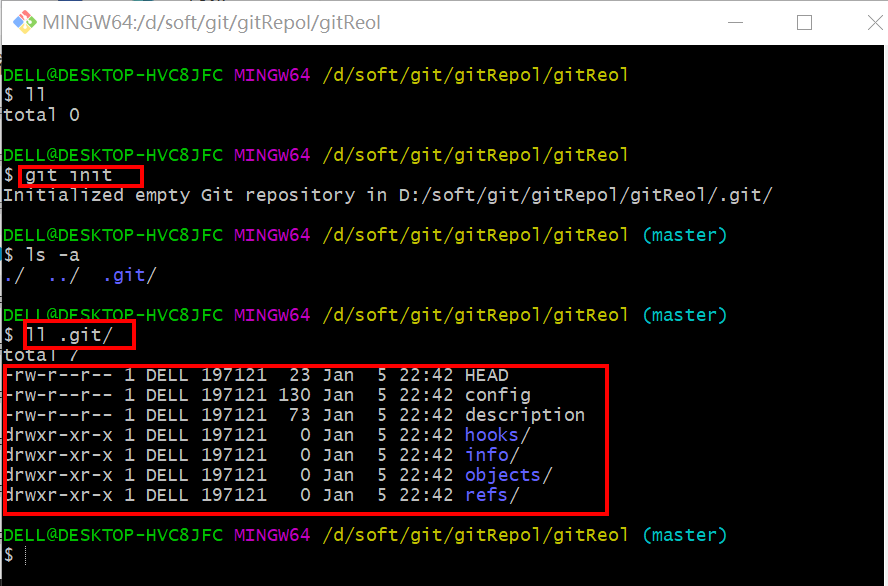


****

# **2 Git 命令行操作**

## 2.1 本地库初始化

* 命令: git init
* 效果：



* 注意：.git目录中存放的是本地库相关的子目录和文件，不要删除，也不要胡乱修改。

## 2.2 设置签名

* 形式

用户名：xiangcm

Email地址：xiangcm163@163.com

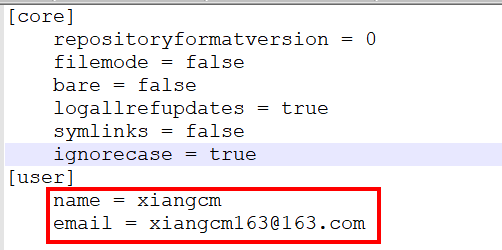
* 作用

区分不同开发人员

* 辨析

这里的签名和登录远程库（代码托管中心）的账号和密码没有任何关系

* 命令
  + 项目级别/仓库级别：仅在当前本地库范围类有效
    - git config user.name xiangcm
    - git config user.email [xiangcm163@163.com](mailto:xiangcm163@163.com)
    - 信息保存的位置：./.git/config文件



* + 系统用户级别：登录当前操作系统的用户范围
    - git config --global user.name xiangcm1
    - git config --global user.email [xiangcm1163@163.com](mailto:xiangcm1163@163.com)
    - 信息保存的位置：~/.gitconfig文件下



* + 级别优先
    - 就近原则：项目级别优先于系统级别
    - 只有系统用户级别，以系统级别的签名为准
    - 不允许二者都不存在

## 2.3 添加提交以及查看

* 命令

添加：git add <file>

提交：git commit <file> （进入vim编辑器，输入提交信息）

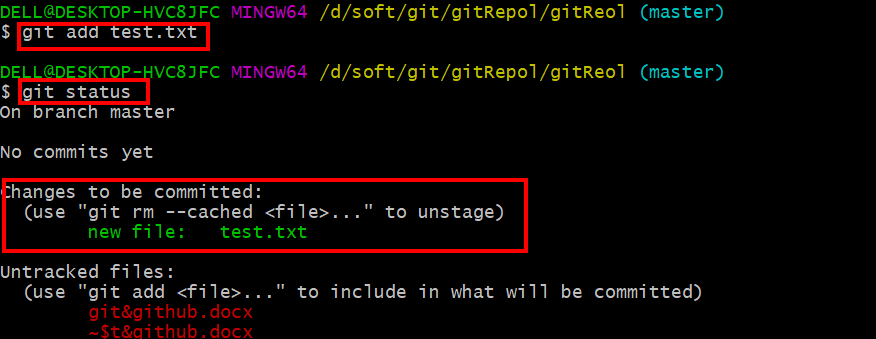
git commit -m <提示信息> <file>

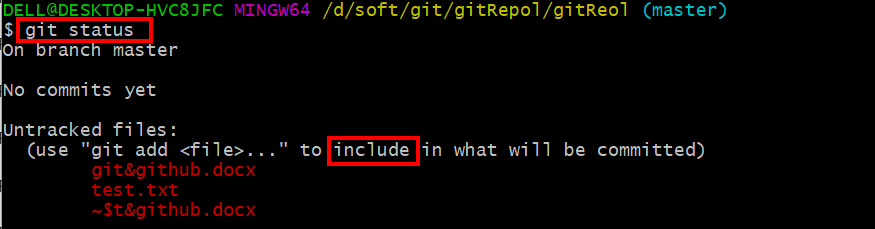
查看：git status

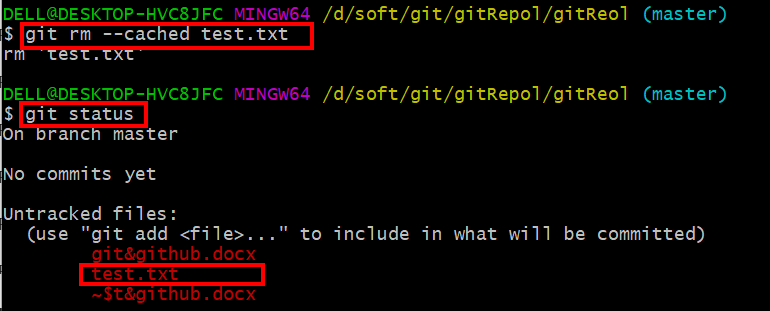
删除在暂存区中的新文件：git rm --cached <file>

删除在暂存区中的修改过的文件：git restore --staged <file>

* 结果



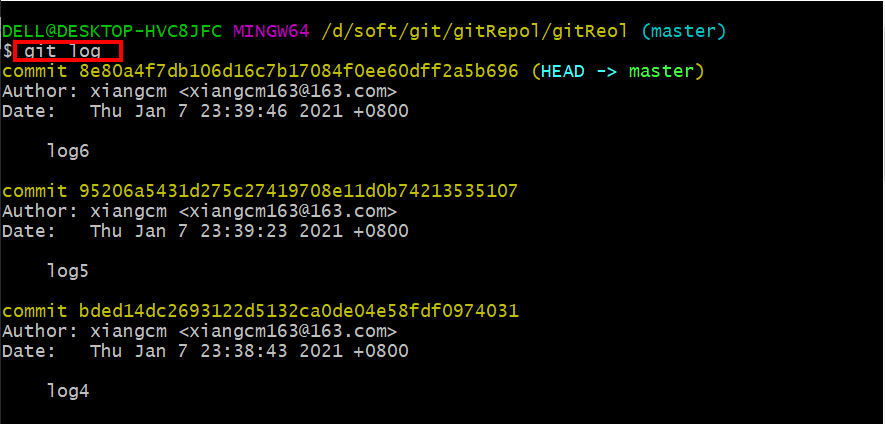


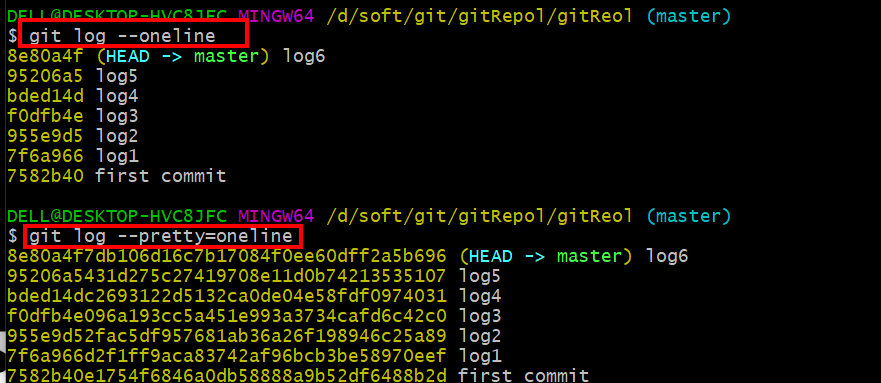


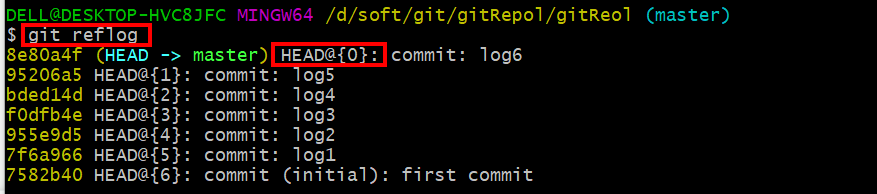


## 2.4 查看历史

* 命令
  + 查看详细提交历史： git log
  + 查看一般提交历史： git log --pretty=oneline
  + 查看简洁提交历史： git log --oneline
  + 显示回退步数提交历史：git reflog
* 结果

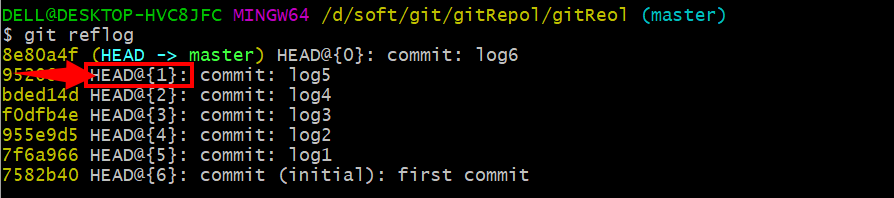




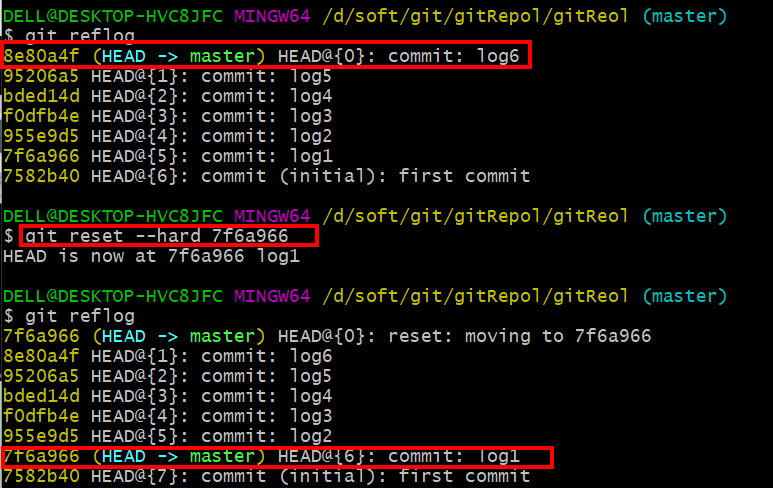


## 2.5 前进操作

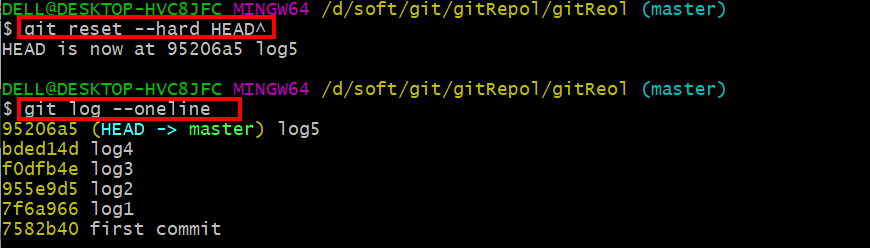
* 本质



* 基于索引值操作
  + 命令：git reset --hard 【局部索引值】（不分前后）
  + 结果：



* 使用^符号：只能后退
  + 命令：git reset --hard HEAD^
  + 注：一个^表示后退一步，n个表示后退n步
  + 结果：

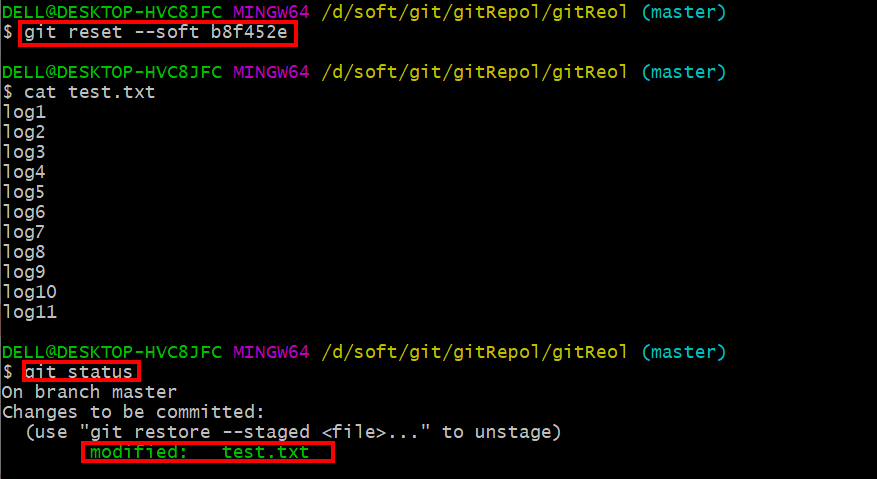


* 使用~符号：只能后退
  + 命令：git reset --hard HEAD~n
  + 注：表示后退n步
  + 结果：



## 2.6 reset 三个参数对比

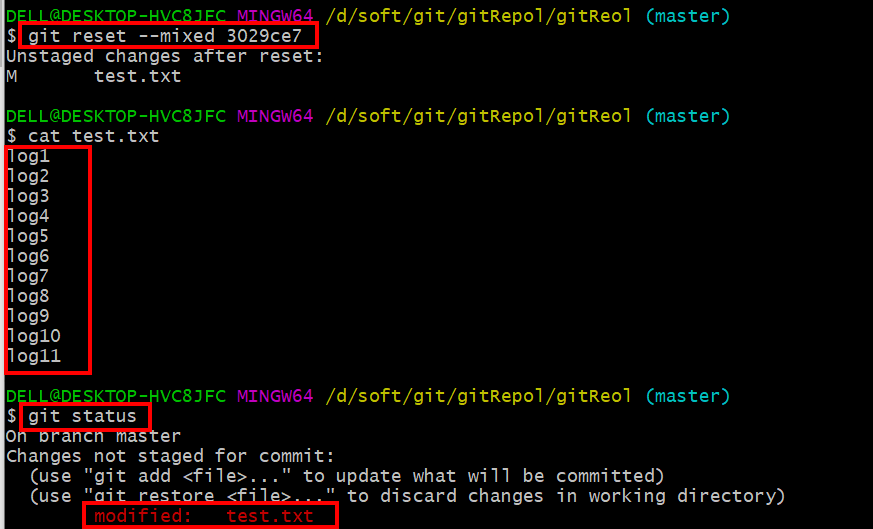
* --soft参数
  + 仅仅子本地库移动HEAD指针
  + 结果：



* + 原因：由于本地库往后退了，相当于暂存区向前一步，所以显得暂存区有了变化



* --mixed参数
  + 在本地库移动HEAD指针
  + 重置暂存区
  + 结果：



* + 原因：由于本地库和暂存区往后退了，相当于工作区向前一步，所以显得工作区有了变化



* --hard参数
  + 在本地库移动HEAD指针
  + 重置暂存区
  + 重置工作区